《数据库》

GaussDB(for MySQL)数据库开发指导手册



华为技术有限公司

目录

[前 言 2](#_Toc61340342)

[简介 2](#_Toc61340343)

[内容描述 2](#_Toc61340344)

[实验环境说明 2](#_Toc61340345)

[实验概览 3](#_Toc61340346)

[1 GaussDB(for MySQL) 数据库开发实验 4](#_Toc61340347)

[1.1 创建用户并授予权限 4](#_Toc61340348)

[1.1.1 新建数据库 4](#_Toc61340349)

[1.1.2 使用DAS创建用户并授予权限 4](#_Toc61340350)

[1.1.3 使用SQL语句创建用户并授予权限 6](#_Toc61340351)

[1.2 创建和管理表 7](#_Toc61340352)

[1.2.1 创建准备 7](#_Toc61340353)

[1.2.2 创建表 7](#_Toc61340354)

[1.2.3 管理表 9](#_Toc61340355)

[1.3 创建和管理其他数据库对象 11](#_Toc61340356)

[1.3.1 创建和管理分区表 11](#_Toc61340357)

[1.3.2 创建和管理索引 16](#_Toc61340358)

[1.3.3 创建和管理视图 17](#_Toc61340359)

[1.3.4 创建和管理存储过程 18](#_Toc61340360)

前 言

简介

本实验指导书主要目的是为了让读者了解GaussDB(for MySQL)数据库的基本操作。

内容描述

本实验指导书通过用户管理、表管理、数据库对象等管理的操作，介绍在DAS环境下如何使用GaussDB(for MySQL)。

实验环境说明

* 组网说明

本实验环境为华为云GaussDB(for MySQL)数据库。如果没有购买GaussDB(for MySQL)则参考《01 GaussDB(for MySQL)数据库实验环境搭建指导手册》进行购买。

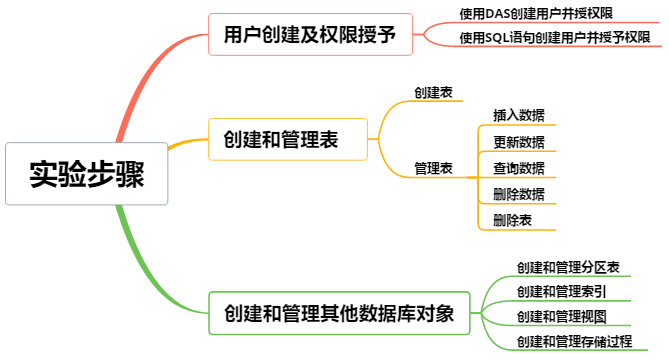
* 设备介绍

为了满足本实验需要，建议实验环境采用以下配置：

设备明细表

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 软件版本 |
| 数据库 | GaussDB(for MySQL) 8 核 | 32 GB | GaussDB(for MySQL)服务 |

实验概览



本实验概览图

# GaussDB(for MySQL) 数据库开发实验

## 创建用户并授予权限

### 新建数据库

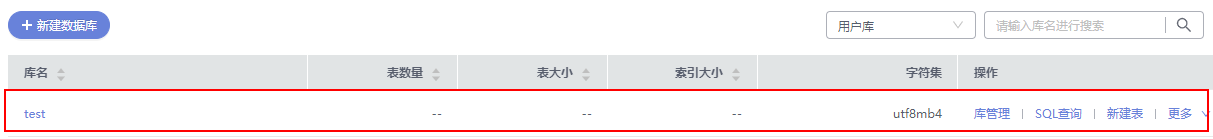
点击“新建数据库”按钮。



输入数据库名称，点击“确定”。

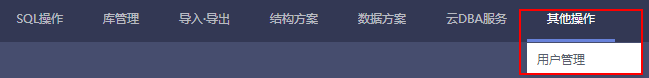


完成数据库创建。



### 使用DAS创建用户并授予权限

在菜单栏中选择“其他操作”选项，并在下拉菜单中选择“用户管理”。

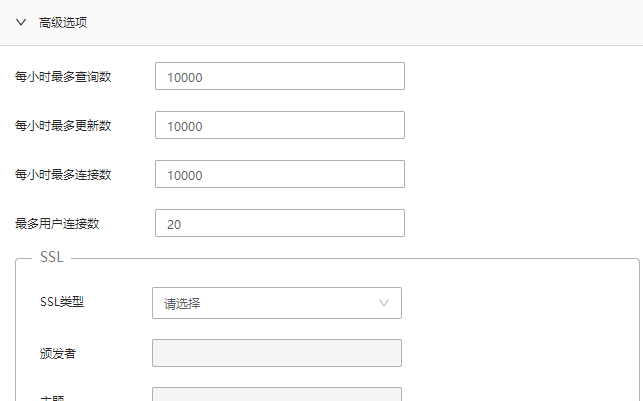


点击“新建用户”，完成“基本信息”、“高级选项”、“全局限制”、“对象权限”参数设置，具体参数根据实际情况而定。

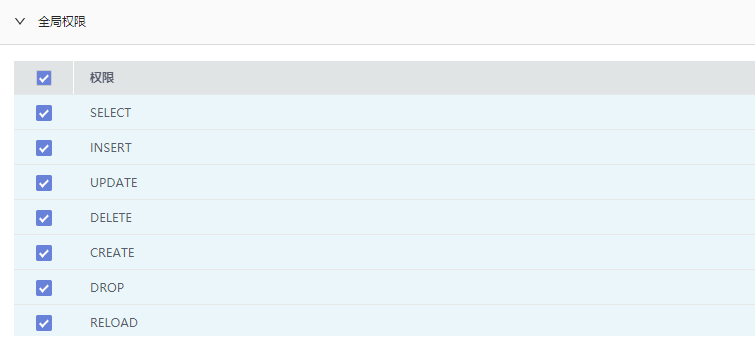
基本信息：



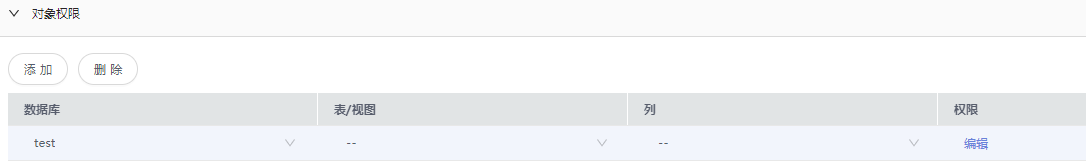
高级选项：



全局权限：



对象权限：点击“添加”，选择数据库对象，这里选择刚刚创建的数据库test。



完成用户创建。



### 使用SQL语句创建用户并授予权限

说明：1.1.2和1.1.3属于两种方式创建用户并授予权限

在“首页”选项卡，点击“SQL查询”，进入SQL查询页界面。





用SQL语句创建，语句如下：

/\*COMMON SETTINGS\*/

**CREATE USER 'user1'@'100.%' IDENTIFIED BY '\*\*\*\*\*\*';**

/\*ADVANCED SETTINGS\*/

**ALTER USER 'user1'@'100.%' WITH**

**MAX\_QUERIES\_PER\_HOUR 10000**

**MAX\_UPDATES\_PER\_HOUR 10000**

**MAX\_CONNECTIONS\_PER\_HOUR 10000**

**MAX\_USER\_CONNECTIONS 20;**

**FLUSH USER\_RESOURCES;**

/\*GLOBAL SETTINGS\*/

**GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, GRANT OPTION ON \*.\* TO 'user1'@'100.%';**

## 创建和管理表

### 创建准备

规划表t\_student内容

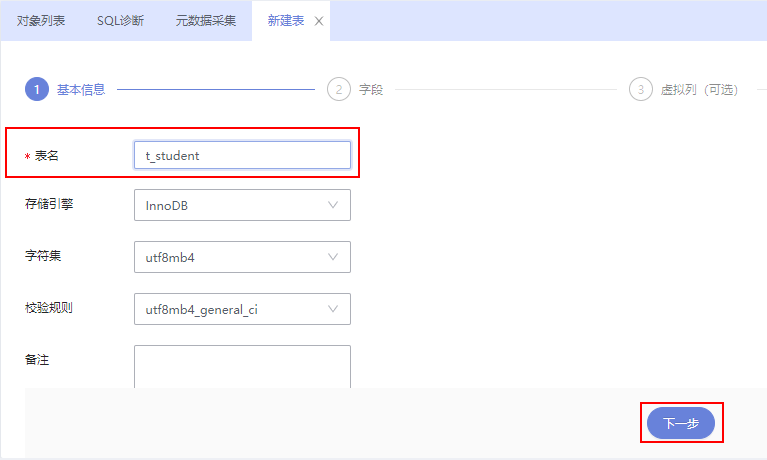
|  |  |  |  |
| --- | --- | --- | --- |
| id | name | age | sex |
| 1 | Jack | 20 | Male |
| 2 | Tom | 21 | Male |
| 3 | Lucy | 22 | Female |

### 创建表

在首页中，选择数据库，在“操作”栏中选择“新建表”。



在“基本信息”中，输入表名，并点击下一步。



在“字段”中，完成字段部分内容填写，并点击“立即创建”。



在“SQL预览”中点击“执行脚本”，完成表创建。



选择“对象列表”，查看创建完的表t\_student，点击“+”号展开。



### 管理表

#### 插入数据

在“SQL查询”页面输入如下语句，完成插入数据。

**INSERT INTO t\_student values(1,'Jack',20,'male'),(2,'Tom',21,'male'),(3,'Lucy',22,'female');**

截图如下：



#### 更新数据

将Jack年龄更改为25，在“SQL查询”页面输入如下语句，完成数据更新。

**update t\_student set age=25 where name='Jack';**

截图如下：



#### 查询数据

查询所有数据，在“SQL查询”页面输入如下语句，完成数据查询。

**select \* from t\_student;**

截图如下：



条件查询，查询id为2的学生信息，在“SQL查询”页面输入如下语句，完成数据查询。

**select \* from t\_student where id=2;**

截图如下：

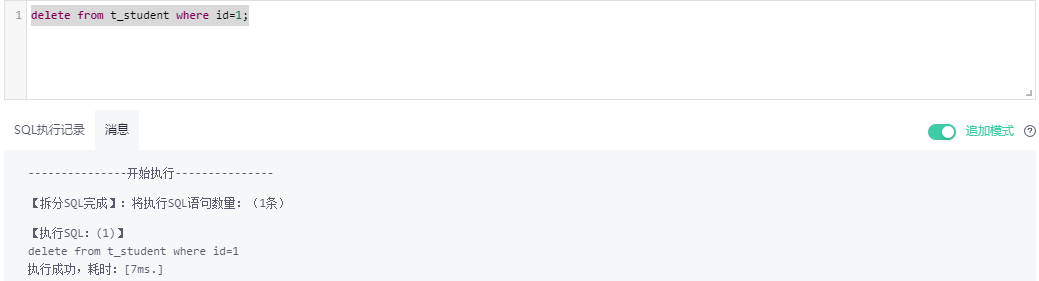


#### 删除数据

删除单条数据，删除id为1的数据，在“SQL查询”页面输入如下语句，完成单条数据删除。

**delete from t\_student where id=1;**

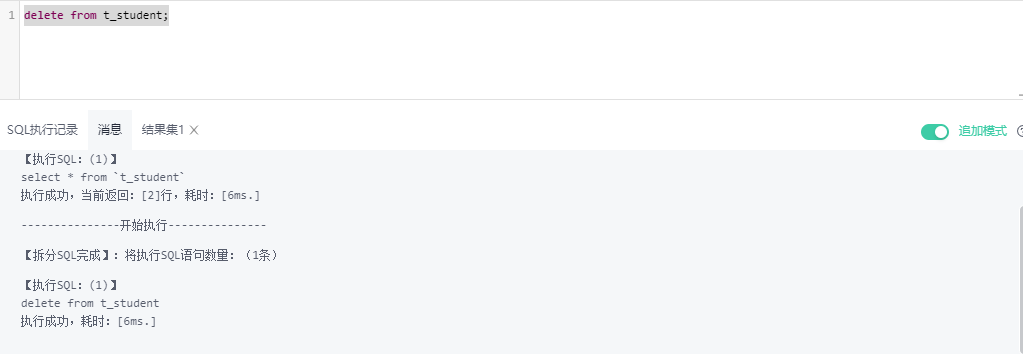
截图如下：



删除所有数据，在“SQL查询”页面输入如下语句，完成单条数据删除。

**delete from t\_student;**

截图如下：



#### 删除表

选择“库管理”界面的“对象列表”，在“操作”栏中点击“更多”，选择“删除”。



在弹出的对话框中选择“确定”，完成表删除。



此外，删除表也可以通过SQL语句执行，语句如下：

**DROP TABLE IF EXISTS t\_student;**

## 创建和管理其他数据库对象

### 创建和管理分区表

删除已存在的重名表，以表“staffs\_p”为例，在“SQL查询”页面输入如下语句：

**DROP TABLE IF EXISTS staffs\_p;**

截图如下：



创建分区表staffs\_p。

**CREATE TABLE staffs\_p**

**(**

**staff\_ID INT not null,**

**staff\_name VARCHAR(10),**

**higest\_degree CHAR(8),**

**phone\_number BIGINT**

**)**

**PARTITION BY RANGE (staff\_ID)**

**(**

**partition P\_050\_BEFORE values less than (50),**

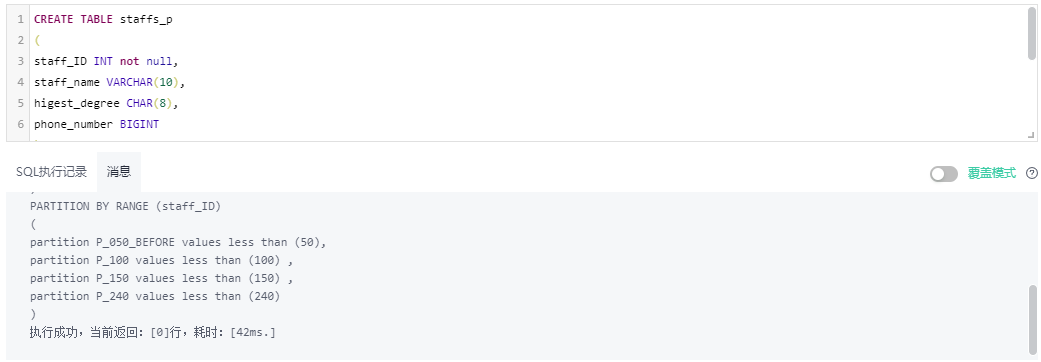
**partition P\_100 values less than (100) ,**

**partition P\_150 values less than (150) ,**

**partition P\_240 values less than (240)**

**);**

截图如下：

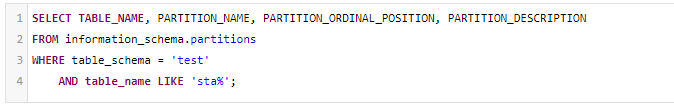


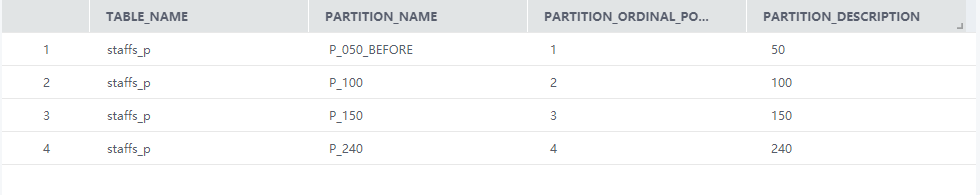
查看表staffs\_p的分区信息。

注意(test为当前自己所访问的库，比如：user01db、user02db等)。

**select TABLE\_NAME, PARTITION\_NAME, PARTITION\_ORDINAL\_POSITION, PARTITION\_DESCRIPTION from information\_schema.partitions where table\_schema='test' and table\_name like 'sta%';**

截图如下：





删除分区P\_150。

**ALTER TABLE staffs\_p DROP PARTITION P\_150;**

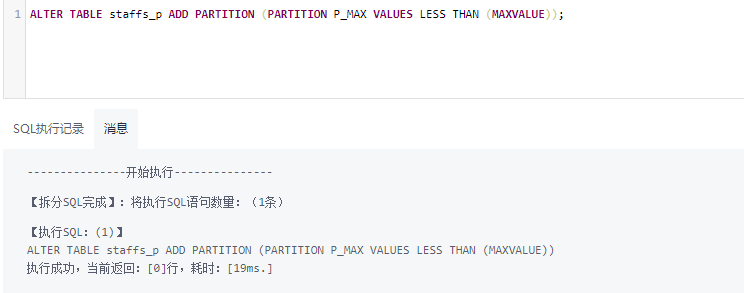
截图如下：



增加分区P\_MAX，范围为 240 <= P\_MAX。

**ALTER TABLE staffs\_p ADD PARTITION (PARTITION P\_MAX VALUES LESS THAN (MAXVALUE));**

截图如下：



分裂名为P\_050\_BEFORE的分区，将其分裂为P\_15和P\_50，其中P\_15的边界是15，P\_50的边界是50。

**ALTER TABLE staffs\_p**

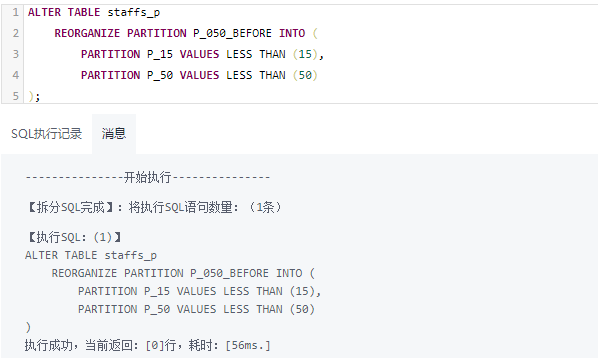
**REORGANIZE PARTITION P\_050\_BEFORE INTO (**

**PARTITION P\_15 VALUES LESS THAN (15),**

**PARTITION P\_50 VALUES LESS THAN (50)**

**);**

截图如下：

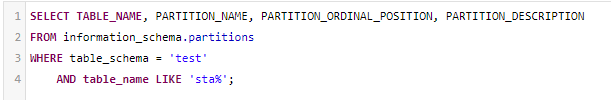


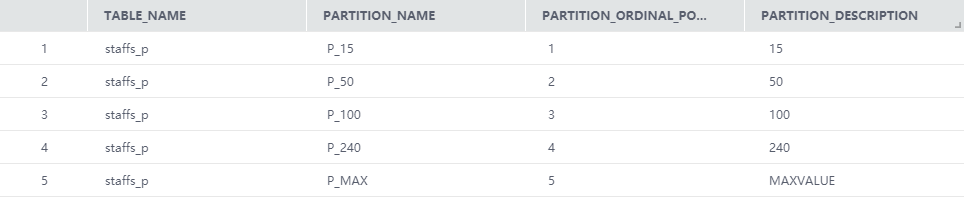
再次查看表staffs\_p的分区信息。

注意(test为当前自己所访问的库，比如：user01db、user02db等)。

**select TABLE\_NAME, PARTITION\_NAME, PARTITION\_ORDINAL\_POSITION, PARTITION\_DESCRIPTION from information\_schema.partitions where table\_schema='test' and table\_name like 'sta%';**

截图如下：





向分区表staffs\_p插入数据。

**insert into staffs\_p values(10, 'tom', 'bachelor', 15200086001);**

**insert into staffs\_p values(20, 'jony', 'master', 13948560111);**

**insert into staffs\_p values(55, 'annie', 'doctor', 15208470001);**

**insert into staffs\_p values(110, 'tom', 'bachelor', 13914086001);**

**insert into staffs\_p values(5, 'rose', 'master', 15208088601);**

**insert into staffs\_p values(25, 'smith', 'master', 13900077001);**

截图如下：



查询分区表staffs\_p数据。

**SELECT \* FROM staffs\_p;**



查询分区P\_15。

**SELECT \* FROM staffs\_p PARTITION(P\_15);**

截图如下：



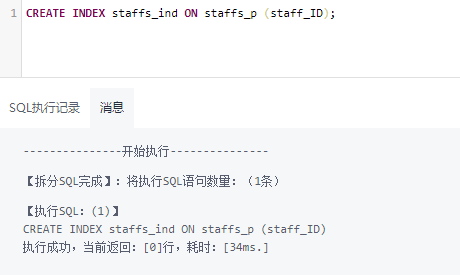
### 创建和管理索引

索引可以提高数据的访问速度，但同时也增加了插入、更新和删除表的处理时间。所以是否要为表增加索引，索引建立在哪些字段上，是创建索引前必须要考虑的问题。需要分析应用程序的业务处理、数据使用、经常被用作查询条件或者被要求排序的字段来确定是否建立索引。

创建索引staffs\_ind。

**CREATE INDEX staffs\_ind ON staffs\_p (staff\_ID);**

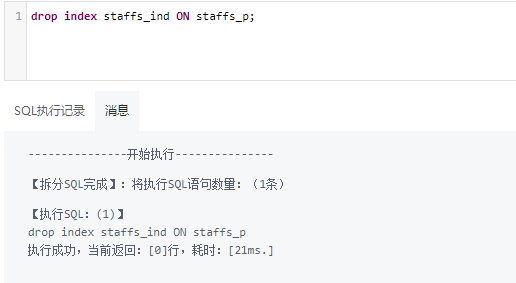
截图如下：



删除索引staffs\_ind。

**drop index staffs\_ind ON staffs\_p;**

截图如下：



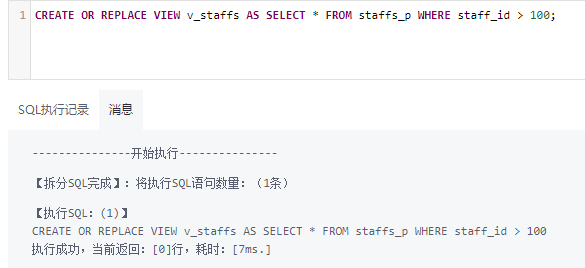
### 创建和管理视图

当用户对数据库中的一张或者多张表的某些字段的组合感兴趣，而又不想每次键入这些查询时，用户就可以定义一个视图，以便解决这个问题。

视图与基本表不同，不是物理上实际存在的，是一个虚表。数据库中仅存放视图的定义，而不存放视图对应的数据，这些数据仍存放在原来的基本表中。若基本表中的数据发生变化，从视图中查询出的数据也随之改变。从这个意义上讲，视图就像一个窗口，透过它可以看到数据库中用户感兴趣的数据及变化。视图每次被引用的时候都会运行一次。

创建视图v\_staffs，若该视图存在则更新该视图。

**CREATE OR REPLACE VIEW v\_staffs AS SELECT \* FROM staffs\_p WHERE staff\_id > 100;**



查询视图中的数据。

**SELECT \* FROM v\_staffs;**

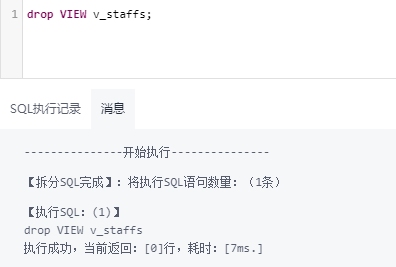
截图如下：



删除视图。

**drop view v\_staffs;**

截图如下：

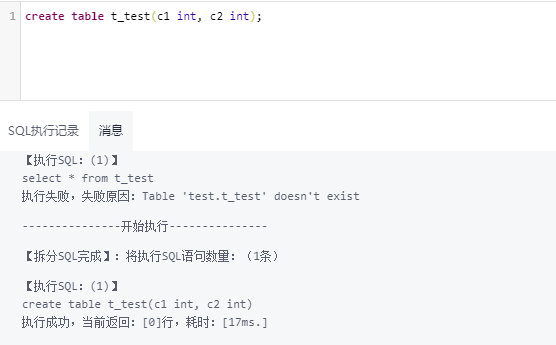


### 创建和管理存储过程

创建表t\_test。

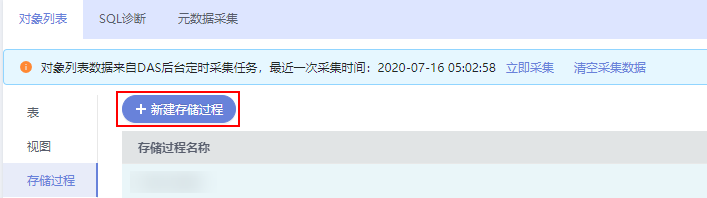
**create table t\_test(c1 int, c2 int);**

截图如下：

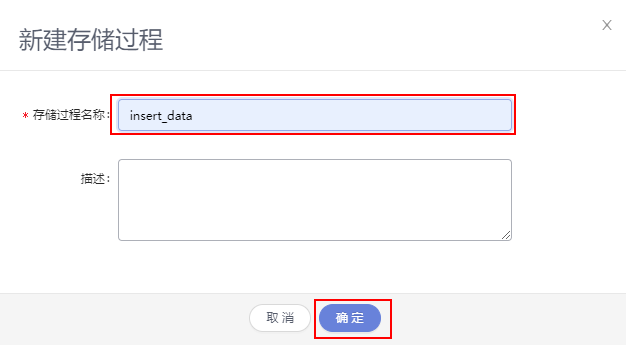


创建存储过程insert\_data。

在“库管理界面”选择“存储过程”，点击“新建存储过程”。



输入存储过程名称，点击“确定”。



在“新建存储过程”中，输入语句，完成后单击“保存”。



存储过程参考代码(test为当前自己所访问的库，比如：user01db、user02db等)：

**CREATE PROCEDURE `test`.`insert\_data`(a int, b int)**

**BEGIN**

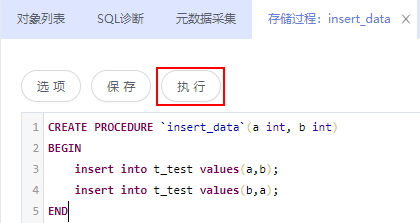
**insert into t\_test values(a,b);**

**insert into t\_test values(b,a);**

**END**

调用存储过程。

在“存储过程：insert\_data”界面点击“执行”。



在“请设置存储过程的入口参数值”对话框中，输入参数a、b的参数值，点击“开始执行”。



SQL参考语句：

**CALL `test`.`insert\_data`(1, 2);**

查询表内容。

**SELECT \* FROM t\_test;**

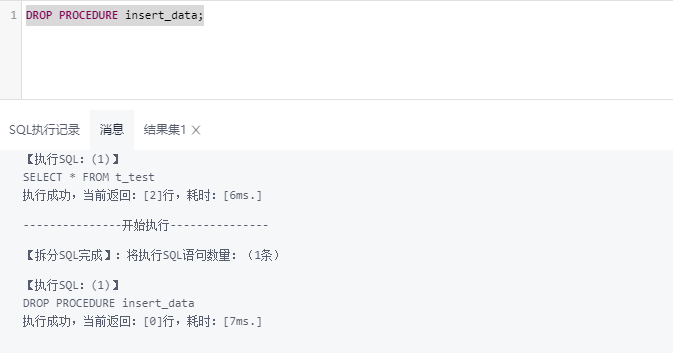
截图如下：



删除存储过程。

**DROP PROCEDURE insert\_data;**

截图如下：



本实验结束。